Principles of software composition 2018/19

Mid-term exam – April 3, 2019

[**Ex. 1**] Suppose we add to IMP the command **repeat** c **until** b, whose denotational semantics is defined recursively as:

$$\mathcal{C}[\![\mathbf{repeat}\ c\ \mathbf{until}\ b]\!]\sigma = (\lambda\sigma'.\mathcal{B}[\![b]\!]\sigma' \ \rightarrow\ \sigma'\ ,\ \mathcal{C}[\![\mathbf{repeat}\ c\ \mathbf{until}\ b]\!]\sigma')^*\ (\mathcal{C}[\![c]\!]\sigma)$$

1. Define the operational semantics of the new construct.
2. Extend the proof of determinacy of the operational semantics taking into account the new construct.
3. Define the function $\Gamma_{c,b}$ such that $\mathcal{C}[\![\mathbf{repeat}\ c\ \mathbf{until}\ b]\!] = \mathit{fix}\ \Gamma_{c,b}$.
4. Compute the denotational semantics of **repeat** $x := x+1$ **until true**.

[**Ex. 2**] Consider the CPO$_\perp$ $\mathcal{D} \stackrel{\text{def}}{=} (\wp(\mathbb{N}), \subseteq)$ and the function $f : \wp(\mathbb{N}) \rightarrow \wp(\mathbb{N})$ such that $f(X) \stackrel{\text{def}}{=} \{y \mid \exists x \in X.\ y \leq x\}$, where $\leq$ is the usual total order on $\mathbb{N}$.

1. Is f monotone?
2. Is f continuous?
3. What is the least fixpoint of f? Does f have other fixpoints?

[**Ex. 3**] Write a Haskell function that takes a list `xs` and returns the list of all pairs (`x`, `n`) such that `x` occurs `n` times in `xs`, preserving the order of appearance. For example, given the input "`hello`" the function must return the list

$$[(\texttt{'h'}, \texttt{1}), (\texttt{'e'}, \texttt{1}), (\texttt{'l'}, \texttt{2}), (\texttt{'o'}, \texttt{1})]$$

[**Ex. 4**] Consider the HOFL terms

$$t_0 \stackrel{\text{def}}{=} \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ (x, f\ x)\ \mathbf{else}\ (f\ x, x)$$
$$t_1 \stackrel{\text{def}}{=} \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ (x, \mathbf{snd}(f\ x))\ \mathbf{else}\ (x, x)$$

Which term is well-typed? What is its principal type?